CLAIMS

1. A method in a computer system for determining resolution of attributes of a program, the method comprising:

providing a program having interactions, each interaction having commands with attributes;

identifying a sequence of interactions of the program; and

for each interaction in the identified sequence,

for each command of the interaction,

for each input attribute of the command,

identifying an output attribute corresponding to the input attribute; and

setting the resolution of the input attribute to the resolution of the identified output attribute; and

for each output attribute of the command, setting the resolution of the output attribute to resolved.

2. The method of claim 1 including reporting input attributes whose resolution is set to unresolved.

3. The method of claim 2 including suppressing the reporting of input attributes that may be resolved by user input.

4. The method of claim 2 including suppressing the reporting of input attributes of primitive types.

5. A method for verifying resolution of input parameters of functions of a computer program before executing the computer program, the method comprising:

providing a path of execution of the computer program, the path of execution identifying a sequence of functions of the computer program; and for each function identified in the provided path of execution, processing the function by

for each input parameter of the function, indicating that the input parameter is resolved when a corresponding output parameter has been indicated as resolved when a function in the path of execution was previously processed; and

for each output parameter of the function, indicating that the output parameter is resolved.

6. The method of claim 5 wherein the computer program is a command-based application wherein the functions are methods of objects corresponding to the commands.

7. The method of claim 6 wherein the commands are organized into interactions.

8. The method of claim 6 wherein the parameters are attributes of the objects.

9. The method of claim 8 wherein the objects have set and get methods for setting and getting attribute values.

10. The method of claim 8 wherein the attribute values are set with an assignment statement.

11. The method of claim 6 wherein each object has a perform method for performing a behavior associated with the command.

12. The method of claim 5 wherein input and output parameters correspond when they have the same name.

13. The method of claim 5 including creating a list of each parameter of each function processed, the list indicating resolution of the parameter, and including outputting an indication of resolution of each parameter based on the created list.

14. The method of claim 5 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands, and wherein each command has a corresponding object with attributes.

15. The method of claim 14 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

16. The method of claim 14 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

17. A computer system for verifying resolution of input parameters of functions of a computer program before executing the computer program, comprising:

means for selecting each function in execution order; and

means for processing each selected function by for each input parameter of the function, indicating that the input parameter is resolved when a corresponding output parameter has been indicated as resolved when a function of the computer program was previously processed and for each output parameter of the function, indicating that the output parameter is resolved.

18. The computer system of claim 17 wherein the computer program is a command-based application wherein the functions are methods of objects corresponding to the commands.

19. The computer system of claim 18 wherein the commands are organized into interactions.

20. The computer system of claim 18 wherein the parameters are attributes of the objects.

21. The computer system of claim 20 wherein the objects have set and get methods for setting and getting attribute values.

22. The computer system of claim 18 wherein each object has a perform method for performing a behavior associated with the command.

23. The computer system of claim 17 wherein input and output parameters correspond when they have the same name.

24. The computer system of claim 17 including means for creating a list of each parameter of each function processed, the list indicating resolution of the parameter, and including means for outputting an indication of resolution of each parameter based on the created list.

25. The computer system of claim 17 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

26. The computer system of claim 25 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

27. The computer system of claim 25 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

28. A computer system for processing each function of a computer program prior to runtime by for each input parameter of the function, determining whether a source of the input parameter would be resolved during execution of the computer program and for

each output parameter of the function, indicating that the output parameter is resolved wherein output parameters are sources of input parameters.

29. The computer system of claim 28 wherein the computer program is a command-based application wherein the functions are methods associated with objects corresponding to the commands.

30. The computer system of claim 29 wherein the commands are organized into interactions.

31. The computer system of claim 29 the parameters are attributes of the objects.

32. The computer system of claim 31 wherein the objects have set and get methods for setting and getting attribute values.

33. The computer system of claim 29 wherein each object has a perform method for performing a behavior associated with the command.

34. The computer system of claim 28 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

35. The computer system of claim 34 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

36. The computer system of claim 34 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.

37. A computer-readable medium containing instructions for controlling a computer system to determine prior to runtime resolution of parameters of functions of a computer program, by a method comprising:

identifying a path of execution of the computer program, the path of execution having of functions of the computer program;

for functions in the identified path of execution,

indicating that an input parameter of the function is resolved when a corresponding output parameter has been indicated as resolved; and

indicating that an output parameter is resolved.

38. The computer-readable medium of claim 37 including indicating that an input parameter of a primitive type may be resolved by a user at runtime.

39. The computer-readable medium of claim 37 wherein the computer program is a command-based application where the functions are methods associated with objects corresponding to the commands.

40. The computer-readable medium of claim 39 wherein the commands are organized into interactions.

41. The computer-readable medium of claim 39 wherein the parameters are attributes of the objects.

42. The computer-readable medium of claim 41 wherein the objects have set and get methods for setting and getting attribute values.

43. The computer-readable medium of claim 37 wherein the computer program is specified by an interaction-based definition, wherein interactions include commands and wherein each command has a corresponding object with attributes.

44. The computer-readable medium of claim 43 wherein each command is defined by a descriptor that optionally provides aliasing for names of attributes.

45. The computer-readable medium of claim 43 wherein each command is defined by a descriptor that optionally provides a constant value for an attribute.